DETAILED ACTION

1. This office action is in response to the applicant's communication received on March 20, 2008.

2. Claims 1-30 are presented for examination.

3. The applicants have not amended, cancelled, or added any new claims.

4. Applicant's arguments with respect to claims 1-30 have been considered but are deemed to be moot in view of the new grounds of rejection.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

6. Claims 1-30 are rejected under 35 U.S.C. 102(a) as being anticipated by Jagadish et al., "TIMBER: A native XML database", The VLDB Journal, December 19, 2002.

7. Regarding Claim 1, Jagadish teaches a native XML database.

The method and associated system for a native XML database as taught or suggested by Jagadish includes:

identifying a set of trees in the collection of tree data structures (Abstract, 'The system is based upon a bulk algebra for manipulating trees, and natively stores XML…;

1 Introduction, '…manipulate sets of trees…'; 1 Introduction, 'sets of trees as input and produce a set of trees as output'; page 277, 3.4 Query processing, 4 Tree algebra), each tree in the set of trees having a same structure (page 275, Example 1, 'XML has a tree structure: elements in the document can be structurally related and these structural relationships are meaningful.'; '…to express tree-based XML data to flat tables in a relational schema.'; '…XML schema often produces a relational schema with many tables. Structural information in the tree-based schema is modeled by joins between tables…'); forming a pattern having the same structure as each tree in the set of trees (page 278, paragraph 4, 'We solve this problem through the use of *pattern trees* to specify…'; page 278, column 2, paragraph 1, 'result is the sub-trees', paragraph 2, 'Conditions other than tag names...', paragraph 3, 'XPath is very popular', and paragraph 4, 'All operators...'); and processing the pattern (page 278, column 2, paragraph 1, 'result is the sub-trees', paragraph 2, 'Conditions other than tag names...', paragraph 3, 'XPath is very popular', and paragraph 4, 'All operators...', page 278, Selection, page 279, Projection and Order).

8. Regarding Claims 2, 9, 17, 20, 24, and 27, Jagadish teaches the pattern is processed in lieu of processing each tree in the set of trees (page 279, Projection and Ordering).

9. Regarding Claims 3, 10, and 28, Jagadish teaches processing the pattern comprises applying a query tree to the pattern (page 281, 5 Query evaluation, 5.1 Physical algebra, page 282, Pattern Tree reuse, page 285, 6 Query optimization, 'Query patterns…').

10. Regarding Claims 4, 14, 21, and 29, Jagadish teaches the pattern comprises applying an extend operation to the pattern (page 282, Pattern Tree reuse, ‘pattern tree extension’).

11. Regarding Claims 5, 15, 22, and 30, Jagadish teaches the pattern comprises applying an intersect operation to the pattern (page 285 6.1 Structural join order selection).

12. Regarding Claims 6, 11, 19, and 26, Jagadish teaches storing the pattern in a computer-readable memory (page 276, 3 System architecture, 3.1 Data Storage); storing the leaf node of each tree in the set of trees in a computer-readable memory (page 276, 3.1 Data Storage, column 2, paragraph 1, ‘For storage efficiency…’); and a leaf node having a value (page 276, column 2, paragraphs 1-3).

13. Regarding Claims 7 and 12, Jagadish teaches the pattern is stored in lieu of storing the same structure of each tree in the set of trees (page 276, 3 System architecture, 3.1 Data Storage’).

14. Regarding Claim 8, Jagadish teaches partitioning the collection of tree data structures into disjoint sets of trees (page 275, Example 1, ‘XML has a tree structure: elements in the document can be structurally related and these structural relationships are meaningful.’; ‘…to express tree-based XML data to flat tables in a relational schema.'; ‘…XML schema often produces a relational schema with many tables. Structural information in the tree-based schema is modeled by joins between tables…’), each set of trees comprising trees of a same structure (page 275, Example 1, ‘XML has a tree structure: elements in the document can be structurally related and these

structural relationships are meaningful.'; '…to express tree-based XML data to flat tables in a relational schema.'; '…XML schema often produces a relational schema with many tables. Structural information in the tree-based schema is modeled by joins between tables…'); forming a set of patterns, each pattern corresponding to one of the set of trees and each pattern having the same structures as its corresponding set of trees (page 278, paragraph 4, 'We solve this problem through the use of *pattern trees* to specify…'; page 278, column 2, paragraph 1, 'result is the sub-trees', paragraph 2, 'Conditions other than tag names...', paragraph 3, 'XPath is very popular', and paragraph 4, 'All operators...'); and processing the pattern (page 278, column 2, paragraph 1, 'result is the sub-trees', paragraph 2, 'Conditions other than tag names...', paragraph 3, 'XPath is very popular', and paragraph 4, 'All operators...', page 278, Selection, page 279, Projection and Order); and processing the set of patterns (page 278, paragraph 4, 'We solve this problem through the use of *pattern trees* to specify…'; page 278, column 2, paragraph 1, 'result is the sub-trees', paragraph 2, 'Conditions other than tag names...', paragraph 3, 'XPath is very popular', and paragraph 4, 'All operators...'); and processing the pattern (page 278, column 2, paragraph 1, 'result is the sub-trees', paragraph 2, 'Conditions other than tag names...', paragraph 3, 'XPath is very popular', and paragraph 4, 'All operators...', page 278, Selection, page 279, Projection and Order).

15. Regarding Claim 13, Jagadish teaches each distributed processor processing one or more of the patterns in the set of patterns (page 278, paragraph 4, 'We solve this problem through the use of *pattern trees* to specify…'; page 278, column 2, paragraph

1, 'result is the sub-trees', paragraph 2, 'Conditions other than tag names...', paragraph 3, 'XPath is very popular', and paragraph 4, 'All operators...'); and processing the pattern (page 278, column 2, paragraph 1, 'result is the sub-trees', paragraph 2, 'Conditions other than tag names...', paragraph 3, 'XPath is very popular', and paragraph 4, 'All operators...', page 278, Selection, page 279, Projection and Order).

16. Regarding Claim 16, the limitations of this claim have been noted in the rejection of claim 1 presented above. In addition, Jagadish teaches a database component operative to maintain a database comprising the collection of tree data structures (page 276, 3 System architecture, 3.1 Data storage, column 2, paragraphs 1 and 2) and processing component communicatively connected to the database component (page 276, 3 System architecture, 3.1 Data storage, column 2, paragraphs 1 and 2, page 277, paragraph 2, “In relational databases…’).

17. Regarding Claims 18 and 25, the limitations of this claim have been noted in the rejection of claim 3 presented above. In addition, Jagadish teaches an input component connected to the processing component (page 276, 3 System architecture, 3.1 Data storage, column 2, paragraphs 1 and 2, page 277, paragraph 2, “In relational databases…’, page 277, Index storage, 3.3 Metadata storage, and 3.4 Query processing) and receiving information from the input component (page 276, 3 System architecture, 3.1 Data storage).

18. Regarding Claim 23, the limitations of this claim have been noted in the rejections of claims 8 and 16 presented above. It is therefore rejected as set forth above.

NAME OF CONTACT

19. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Cheryl Lewis whose telephone number is (571) 272-4113. The examiner can normally be reached on 6:30-3:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on (571) 272-7079. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

(571) 273-4113 (Use this FAX #, only after approval by Examiner, for "INFORMAL" or "DRAFT" communication. Examiners may request that a formal paper/amendment be faxed directly to them on occasions.).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist/ Technology Center (571) 272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov.

Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Cheryl Lewis/
Primary Examiner, Art Unit 2167
June 9, 2008